MINERALS

ECONOMIC & REGIONAL DEVELOPMENT AGREEMENT

Final Report
TASK 4 - Geotechnical Database
for Ground Control
Numerical Modelling
BEAP-DD USER DOCUMENTATION

by

Inco Limited, Thompson Division
Thompson, Manitoba

March 1990

Canada

Manitoba

MG-13097A

CANADA - MANITOBA

MINERAL DEVELOPMENT AGREEMENT

RESEARCH & DEVELOPMENT IN MANITOBA MINES

Work on the project was conducted under the auspices of Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada under the Canada/Manitoba Mineral Development Agreement.

DSS File NO. 14SQ.23440-4-9147-4
DSS Contract Serial No. OSQ84.00404
CANMET Project NO. 4-9147-4
CANMET Scientific Authority: Y.S. Yu

Final Report
TASK 4 - Geotechnical Database
for Ground Control
Numerical Modelling
BEAP-DD USER DOCUMENTATION

by

Inco Limited, Thompson Division
Thompson, Manitoba

March 1990

MICROMEDIA

14 pp

90-05658/1

CANADA - MANITOBA

MINERAL DEVELOPMENT AGREEMENT

RESEARCH & DEVELOPMENT IN MANITOBA MINES

GEOTECHNICAL DATABASE FOR GROUND CONTROL

Numerical Modelling

Beap-DD User Documentation

This has been prepared as part of the

TASK 4
Geotechnical Database for Ground Control
of the
Canada/Manitoba Mineral Development Agreement

by

J.A.C. (Tony) Diering
Steffen, Robertson, & Kirsten (B.C.) Inc.
Sub-contracting to
Inco Limited, Thompson Division, Manitoba

DSS File No.: 14SQ.23440-4-9147-4
CANMET Project No.: 4-9147-4
CANMET Scientific Authority: Y.S. Yu

USER DOCUMENTATION

FOR

BEAP-DD : VERSION 1.0

BOUNDARY ELEMENT ANALYSIS PACKAGE

DISPLACEMENT DISCONTINUITY MODULE

VOLUME 1 OF 1 : USER MANUAL

REF 54703 NOVEMBER 1989

TABLE OF CONTENTS

			Page
1.0	OVERVIEW		1
	1.1	APPLICATION AREAS	1
	1.2	FEATURES	1
	1.3	LIMITATIONS	2
	1.4	ABOUT THIS MANUAL	2
	1.5	GEOMETRIC NODES, LUMPS AND ELEMENTS.	2
2	INSTALLATION		5
	2.1	GETTING STARTED	5
	2.2	DATABASE FILES	6
	2.3	PROGRAM FILES	7
	2.4	TEST DATA FILES	8
3	PREPARATION OF USER INPUT FILES		9
	3.1	D201 GLOBAL GEOMETRY (NODES AND LUMPS)	10
	3.2	D202 LUMP GROUPING	12
	3.3	D203 LUMP MATERIAL PROPERTIES	14
	3.4	D207 BOUNDARY CONDITIONS, BENCHMARKS AND GENERAL INPUT	16
	3.5	KEYWORDS FOR D207	17
	3.6	D503 DEFINE BACKGROUND LAYOUT	31
	3.7	D505 CHANGE LAYOUT FOR A MINING STEP	32
	3.8	D515 READ IN A MINE LAYOUT AS AN ARRAY	33
	3.9	CODES FOR ELEMENT TYPE DEFINITION	35
	3.10	MODELLING THE EARTH'S SURFACE	41

		Page
4	DESCRIPTION OF PROGRAM MODULES	43
5	RUNNING BEAP	58
	5.1 EXAMPLE 1	58
	5.2 EXAMPLE 2	60
	5.3 EXAMPLE 3	62
6	REFERENCES	64
	APENDIX 1 -- SAMPLE DATA FILES.	64

LIST OF FIGURES

Figure		Page
1.5.1	Local and intrinsic co-ordinates for a single lump, also showing element numbering.	3
1.5.2	Geometry for small test example.	4
3.10.1	Various lump configurations to model earth's surface	42

1.0 AN OVERVIEW

BEAP-DD facilitates three-dimensional stress analysis of underground excavations which are tabular in nature. It is an interactive package which has been designed and written specifically for application to mining problems. Excavation surfaces and sometimes the earth's surface are modelled with displacement discontinuity elements. The element type used is a square displacement discontinuity element with constant variation of closures and tractions across each element.

The package evolved from the boundary element package MBEM which already has been used or installed in many countries. BEAP-DD is part of a system of programs BEAP and BEAP-M and will often be used with those programs.

1.1 APPLICATION AREAS

- Analysis of single seam or reef mine layouts.
- Pillar design for tabular deposits
- Analysis of a single reef or seam close to a fault plane
- Analysis of a folded seam or reef
- Analysis of multiple seams or reefs.

1.2 FEATURES

- IBM-PC/AT or compatibles
- Interactive or batch mode operation
- Uses user digitized polygons for layout definition
- Multi step capability
- Capacity of up to 256 lumps (6400 elements) or 19200 degrees of freedom
- Named elements and benchmarks for easy user identification
- Use of iterative solution technique and lumping to reduce computing requirements
- Robust against user abuse with various re-start options
- Includes fault or non-linear backfill elements.
- Fully compatible with BEAP and BEAP-M

1.3 LIMITATIONS

- Currently set for a maximum of 100 X 100 elements for a single seam or less if multiple seams are used.

- Large problems may require long execution times. On IBM-PC type equipment (even using 80386 processors), overnight runs will usually be required. Disk space requirements will vary depending upon problem definition, but may be large in some instances.

- This manual has been prepared for the assistance and guidance of users of BEAP-DD. Whilst care has been taken in the presentation of this manual, the user should treat results with due caution and engineering judgement. By the very nature of such programs, an in-experienced user can easily obtain erroneous results. Moreover, as there so many different types of problems which can potentially be analyzed by such a program, it is possible (and more likely probable that there will be program errors in some circumstances). Such errors can only be ironed after continued use and debugging of the program.

1.4 ABOUT THIS MANUAL

BEAP-DD enables advanced elastic stress analysis of fully three-dimensional mining geometries to be carried out efficiently and economically.

This manual assumes that the user is already familiar with the concepts of three-dimensional stress analysis, particularly the use of displacement discontinuity elements for such modelling, and the associated concepts of working with a three-dimensional geometry. For further information in this regard, the user is referred to the list of references listed at the end of the manual.

Section 2 of this manual describes installation procedures, while Section 3 describes the preparation of the input data files Section 4 describes each of the program modules making up the BEAP-DD system. Procedures for running BEAP-DD are given in Section 5.

Less experienced users are well advised to read through the papers listed under References in Section 6.

1.5 GEOMETRIC NODES, LUMPS AND ELEMENTS.

Nodes are points in space. A node is uniquely defined by a coordinate location (x, y, z). In BEAP-DD a LUMP is a collection of 5 X 5 displacement discontinuity elements each of which is assumed to be square. The location in space and size and shape of each lump depend upon the location of the nodes defining that element.

Geometric nodes are used to define the location of the element in space. 4 geometric nodes are required for each lump. Once these geometric nodes have been used to define the lump locations, the program then generates and uses the elements within each lump.

Figure 1.5.1 shows the internal arrangement of elements within a single lump. Note that local row and column numbers correspond with local X and Y coordinate axes respectively. Figure 1.5.2 shows the arrangement of nodes and lumps for a small test example. The data files for this test example are also included as an appendix.

Figure 1.5.1 shows an internal arrangement of nodes within a single boundary element. Figure 1.5.2 shows the internal arrangement of elements within a single displacement discontinuity lump. Note that local row and column numbers correspond with local X and Y coordinate axes respectively.

Often, in this manual, BE is used to denote boundary elements, whilst DD is used to denote displacement discontinuity lumps and/or elements.

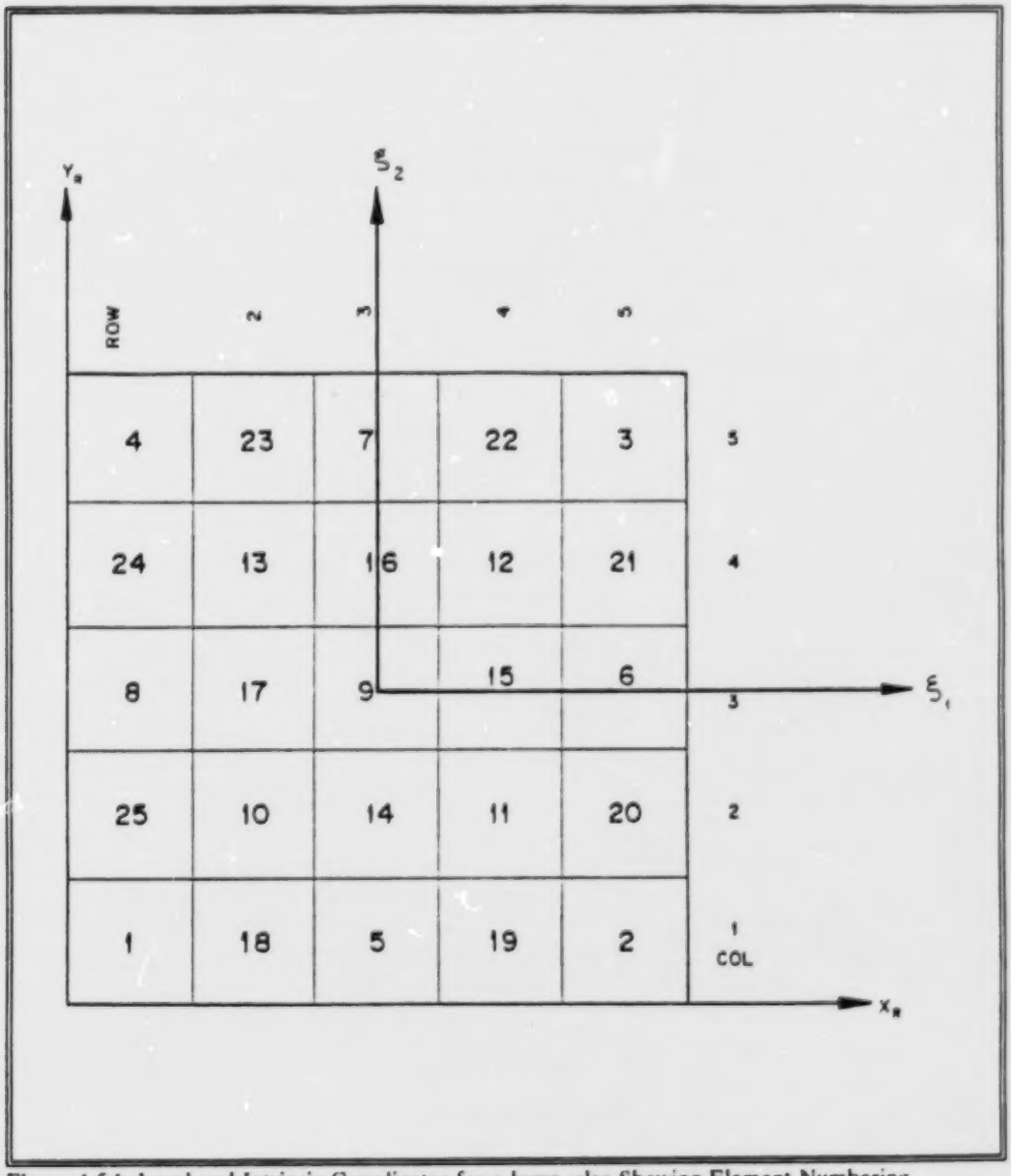

Figure 1.5.1: Local and Intrinsic Coordinates for a lump, also Showing Element Numbering

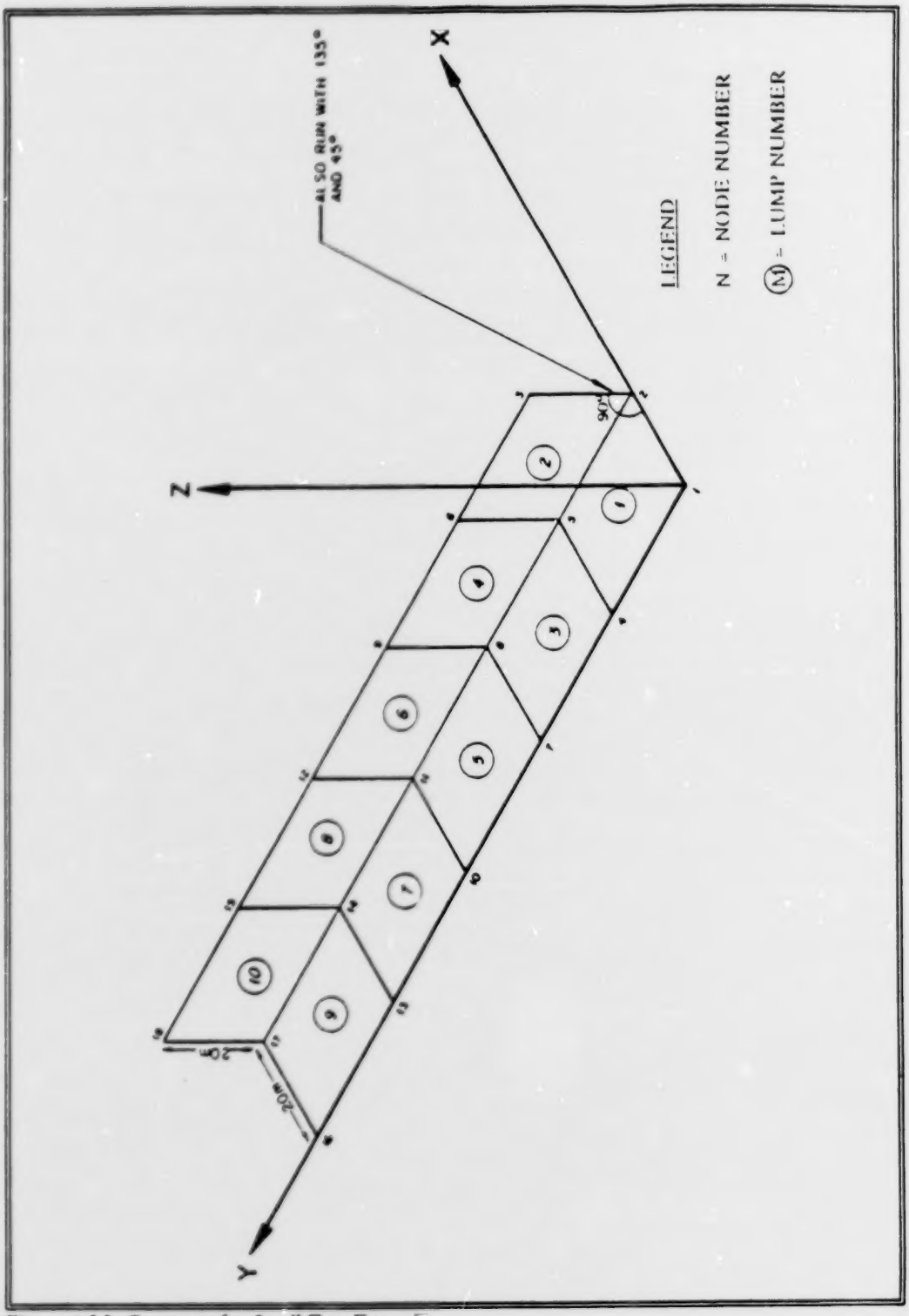

Figure 1.5.2: Geometry for Small Test Example

2.0 INSTALLATION

This installation procedure described here applies only to IBM-PC compatible equipment running DOS. BEAP-DD has, however, already also been installed successfully on an IBM mainframe computer under the VM/CMS operating system.

2.1 GETTING STARTED

It is advisable to create a separate subdirectory which will contain the BEAP-DD program modules and a second subdirectory to contain the data files and database. If BEAP-DD is being installed along with BEAP and / or BEAP-M, then the program subdirectory may be the same as that containing the other program modules. A sound working knowledge of DOS is required for a successful installation of BEAP-DD. Let us call the program subdirectory 'BEAP' and assume that it is on the C Drive. Let us call the database subdirectory 'BEAPDATA' and assume that it is on the D Drive. If there is insufficient space on the C drive to install all of the executable code for BEAP-DD, then it is still necessary to create a subdirectory C:\BEAP which MUST contain the message files for BEAP-DD (and also BEMPOST, BEAP and BEAP-M if these are also installed). The AUTOEXEC.BAT must be modified so that the BEAP directory which contains the executable code (ie all the .EXE files) is included in the DOS PATH. The CONFIG.SYS file must also be modified to ensure that the 'FILES = 20' statement is included.

The program modules occupy around 4 Megabytes of disk space. The test database supplied with BEAP-DD is very small (less than 1 MByte). However, storage requirements for the database can be very large. At least 15 Megabytes should be available for medium-sized problems with 30 Megabytes or more for larger problems.

2.2 DATABASE FILES

The user is required to create several input data files. The exact file names can be defined by the user. However, it may often be useful to use the 3 character file type identifier of each data file to indicate the program module which will be processing that file. For example 'twoseam.203' might be a data file which will be processed by module D203. When a new database is created using M101, numerous other data files will be created in the current subdirectory from which the program is being run. Thus, it is possible to run the program from a separate sub-directory from that which contains the data input files, although this is usually not necessary.

Most of the database files are in binary code and cannot be usefully displayed by the user. There are program modules for decoding or displaying these files where necessary.

Database files are as follows:

BEMPOL.	BE	Polygon master file
BENSTOR.	BE	Benchmark file (Binary)
BOTH.	BE	Basic information (internal to program)
VARCOM.	BE	Basic information (to supplement BOTH.BE)
XN.	BE	Nodal coordinates
TEMP.	LST	printout file for results
BEMCO.	DD	Influence coefficients (BE)
BEMMESH.	DD	Plot file for DD elements
BEMMESHE.	DD	Plot file of DD element numbers
BEMMESHN.	DD	Plot file of DD node locations (as X)
BEMSTEP.	DD	Store information for multi step runs
BVECT.	DD	Right hand side vector
LIND.	DD	Lumping maps (BE-DD)
LUMPS.	DD	Geometric information for DD lumps
LUMSTOR.	DD	Result information for DD elements
LUMSTOR2.	DD	More information for DD elements
LUMSTORG.	DD	Geometric information for DD elements
POLPLOT.	DD	Plot file for polygons
SELFCO.	DD	Self effects for DD elements
SHEET.	DD	File to store spreadsheet information
STORSTEP.	DD	Store information for multi step runs

2.3 PROGRAM FILES

The following .EXE files make up the BEAP-DD system. Programs may be run independently, or via the menu driver BEAPDD.EXE.

The following program modules make up the BEAP-DD system :-

BEAPDD	Menu driver for BEAP-DD (and parts of BEAP-M)
M101	Initialize an entire new database (BEAP and BEAP-DD)
D102	Initialize polygon master file
D103	Initialize closures and stresses
D201	Read in node and DD lump geometry (file D201)
D202	Read in lump grouping (file D202)
D203	Read in lump material properties (file D203)
D204	Print selected DD lumps (summary)
D205	Print selected DD lumps (detailed)
D206	Print out summary of polygons
D207	Read in remaining input (file D207)
D208	Print header information
D209	Print lump summary information (one line per lump)
D301	Generate a grid of lumps
D401	Prepare 3-D view of lump geometry
D402	Prepare polygons for plotting
D501	Calculate lumping maps (DD-DD)
D502	Calculate influence coefficients (DD-DD)
D503	Initialize layout codes prior to first step (D503)
D504	Increment mining step number
D505	Update layout codes for current step (D505)
D506	Calculate tractions due to primitive stresses
D507	Solve equations (DD)
D508	Calculate benchmark stresses and displacements
D509	Calculate energy release rates
D510	Store element results for future reference
D511	Store benchmark values for future reference
D515	Read in a mine layout (in array format) for current step
D601	Define current step number
D602	Re-load element results
D603	Re-load benchmark results
D604	Print selected element results
D605	Print current layout as a grid
D606	Print selected benchmark results
D607	Write out selected results as PC-MINE extract file
D608	Write out selected results in spreadsheet format

D701	Prepare element data for plotting
D702	Prepare benchmarks for plotting (BEMPOST)
BEMPOST	Post processor for benchmarks
D800	Digitizer utilities (separate menu)
D809	Generate a test circular polygon ???
D901	Print global nodes and lumps (data file format)
D903	Display/Modify number of nodes/lumps/benchmarks
D904	Set automatic input files on/off
D905	Load in a new message file
D906	Access/display selected messages
D907	Print out selected messages
D908	Set compact of full print mode

At the time of installation, all these .EXE files would be copied into the program subdirectory (for example C:\BEAP)

2.4 TEST DATA FILES

A number of test data files have been included at the end of this manual to help new users become familiar with BEAP. The data files and examples are as follows:

DREEF — dipping reef close to surface

FAULT — vertical fault intersecting a reef. In this example, only the upper portion of the fault was modelled. This is slightly unrealistic, but keeps the problem size small for testing purposes. (See figure 1.5.2).

Any user running BEAP-DD for the first time should ensure that the program can be made to run with one of these data files (which most closely matches has own problem) before attempting to run BEAP with the actual problem data.

3.0 PREPARATION OF USER INPUT FILES FOR BEAP-DD

The input data for program BEAP-DD consist of various data files. Input data files may be read in interactively or in batch mode. If menus are switched off, then batch or command mode is assumed. In command mode, filenames are assumed the "same" as the command, except that the first letter M is replaced with a D to designate "data". For example, module D201 reads in global nodes and lumps. This file may have any name if the user is running interactively. In command mode, however, the data file name for D201 must be D201. Similarly, the data file name for D202 must be D202, when in command mode. The next section details the structure and format of each data file.

Datafiles are as follows:

D201	Global geometry (nodes and lumps)
D202	Lump grouping
D203	Lump material properties
D207	Boundary conditions, benchmarks and general input
D503	Mining layout control for step "zero"
D505	Mining layout control for subsequent steps
D515	Read in a mine layout as an array

3.1 D201 -- GLOBAL NODES AND ELEMENT DEFINITIONS

This file contains 5 different types of line as follows:

Line 1 NOD,NLUM

Line 2 XOF,YOF,ZOF

Line 3 N,X,Y,Z

Line 4a I,M,I1,I2,I3,I4,I5,I6,I7,I8,I9

Line 4b DESCRIP

where

NOD	is the number of global nodes to be defined. This is equal to the number of lines 3 which follow. Max value for NOD is 400.
NLUM	is the number of lumps to be defined. This is equal to the number of line pairs (4a and 4b) which follow. Max value is 256.
XOF,YOF,ZOF	are coordinate offsets used when global nodes are remote from the coordinate origin. Usually use 0,0,0 here.
N	is a node number (between 1 and 400).
X,Y,Z	are the x,y,z coordinates of that node.
I	is a lump number (between 1 and 256).
M	is the number of nodes for that lump. A value of 4 is recommended for square or rectangular lumps. If M=4, then it is not necessary to define I5,I6,I7,I8 and I9.
DESCRIP	is a description of the lump consisting of up to 16 characters. This description is purely for user convenience, helping identify lumps.

Notes

1 Usually, only four nodes will be used to define a lump. When defining an excavation, the recommended sense of node numbering is such that nodes 1 to 4 are in the order top left, bottom left, bottom right and top right.

2. It is not necessary for all lump numbers to be defined. For example it is possible to allocate lump numbers 1-20 for the first grid or seam, 30-50 for the second and so on. In this case, lumps 21-29 would be undefined, but this is not a problem, so long as the correct definitions are also contained in the other files described below.

3. It is also not necessary to define all geometric nodes. ie missing node numbers are not a problem.

An example of this data file (D201) follows :

This files defines nodes and elements for two intersecting reef planes or seams.

```
18,10
0,0,0
1,0,0,0
2,20,0,0
3,20,0,20
4,0,20,0
5,20,20,0
6,20,20,20
7,0,40,0
8,20,40,0
9,20,40,20
10,0,60,0
11,20,60,0
12,20,60,20
13,0,80,0
14,20,80,0
15,20,80,20
16,0,100,0
17,20,100,0
18,20,100,20
1,4,1,2,5,4
BASE EDGE
2,4,2,3,6,5
SIDE EDGE
3,4,4,5,8,7
BASE RIGHT
4,4,5,6,9,8
SIDE RIGHT
5,4,7,8,11,10
BASE CENTRE
6,4,8,9,12,11
SIDE CENTRE
7,4,10,11,14,13
BASE LEFT
7,4,11,12,15,14
SIDE LEFT
9,4,13,14,17,16
BASE EDGE
10,4,14,15,18,17
SIDE EDGE
```

3.2 D202 -- LUMP GROUPING

This file is used to define the following items for selected lumps.

Variable	Default	Description
IEX	1	"Excavation" number or plane number to which the lump belongs. This is used to separate different sets of lumps which have different sizes or orientations. It is also used to identify the node number (from D201) which defines the origin of the local coordinate system for that set of lumps
IKOD	1	Lump status code (O = Dead) (1 = Normal) (2 = Reveiver only (ie Benchmark)) (3 = Emitter only (ie frozen)
IPEN	1	Pen number for plotting the lump
IMOD	1	Subregion number containing the lump
IGROUP	1	Group number to which the lump belongs. Usually set equal to IEX. IGROUP is used to identify groups of lumps which may be mined separately, or which may be treated separately during a mining sequence
LIBKOD	3	Library code for the lump (use 3)
KODMIN	1	Element code - mined, solid etc - see separate section
ISP2	0	Spare variable -- use 0

The format each line of this data file is the same and is as follows:

LB,LE,IEX,IKOD,IPEN,IMOD,IGROUP,LIBKOD,KODMIN,ISP2

where

LB is the first lump number of a series of lumps.

LE is the last lump number of a series of lumps.

Other variables are as defined above.

The last line of this file is

0,0,0,0,0,0,0,0,0,0

An example of this data file for a single deformable reef plane is

1,999,1,1,1,1,1,3,2,0	All elements have same values
0,0,0,0,0,0,0,0,0,0	End of file marker

A second example of this datafile for two intersecting planes is:

```
1,1,1,1,1,1,1,3,0,0
2,2,2,1,1,1,2,3,0,0
3,3,1,1,1,1,1,3,0,0
4,4,2,1,1,1,2,3,0,0
5,5,1,1,1,1,1,3,0,0
6,6,2,1,1,1,2,3,0,0
7,7,1,1,1,1,1,3,0,0
8,8,2,1,1,1,2,3,0,0
9,9,1,1,1,1,1,3,0,0
10,10,2,1,1,1,2,3,0,0
0,0,0,0,0,0,0,0,0,0
```

3.3 D203 -- LUMP MATERIAL PROPERTIES

This file is used to define material properties for selected lumps. Variables are as follows:

THICK	Seam/reef/fault thickness
UCS	Uniaxial compressive strength of material within the lump
HM	M value for application of Hoek's failure criterion
HS	S value for application of Hoek's failure criterion
EXT	Extension strain limit for surrounding rock
ESEAM	Normal modulus of seam/reef material
GSEAM	Shear modulus of seam/reef material
COHES	Cohesion of fault material
PHI	Angle of friction of fault material
TENS	Tensile strength of fault material
FILL1	First backfill parameter (see backfill documentation)
FILL2	Second backfill parameter (see backfill documentation)
FILL3	Third backfill parameter (see backfill documentation)
FILL4	Fourth backfill parameter (see backfill documentation)
FILL5	Fifth backfill parameter (see backfill documentation)

Each sequence of lump requires two lines of data, the format of which is as follows:

LB,LE,THICK,UCS,HM,HS,EXT
ESEAM,GSEAM,COHES,PHI,TENS,FILL1,FILL2,FILL3,FILL4,FILL5

where

LB,LE are as for file D202

The last line of this file is

0,0,0,0,0,0,0

Notes :

1 Usage of the above variables/properties varies depending upon the element types encountered in each lump. For detailed use of the above variables, see the section on types of element available (eg mined, solid etc.)

An example of this data file for two intersecting planes of lumps is given below.

```
1,1,5,0,85,1,.22,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
2,2,0,1.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
3,3,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25.5,.1,.5,50,2,.02,.1
4,4,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
5,5,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25.5,.1,.5,50,2,.02,.1
6,6,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,1
7,7,5.0,85,1,.2,.002,MINED ELEMENTS
```

Note

1 The user descriptions at the end of each first line are for user convenience only and are not read by the program

3.4 D207 -- REMAINING INPUT DATA

This data file looks after the remaining input data. It caters for a wide variety of data types and these data types are specified using keywords. The order in which keywords is specified not critical, but once a keyword has been specified, the structure and format of data which follows is specific to that keyword.

Often, the user will prepare several different data files making use of these keywords. For example, it is possible to have one such file whose sole function is to change the number of decimal places used for the generation of plot files, and a second such file which will change the integration accuracy parameters and so on.

Valid keywords are as follows:

HEADING	Description of the run
PRIM0	Constant primitive stresses
PRIM1	Overburden (or depth dependent) primitive stresses
BENCH	Define benchmark locations
ITERAT	Iteration information for solution of equations
INTEGR	Integration data for calculation of influence coefficients
SPTRAC	Apply special tractions at selected lumps/elements
ISSW	Set integer control switches
SSW	Set real control switches
SUBS	Define material properties for each subregion
END	Define the end of file

For a 'standard' mining problem, one would use the following keywords :

HEADING
PRIM0
PRIM1
BENCH
SUBS
END

Each keyword is explained separately in the following sections.

3.5 KEYWORDS FOR D207

Keyword **BENCH**

Purpose : To define benchmark locations

Format :

IDB,XBEN,YBEN,ZBEN,ISUB
NAMEB

Variables:

IDB	is an integer identifier. Any value between 1 and 9999 may be used for user convenience
XBEN,YBEN,ZBEN	are the coordinates of the point
ISUB	is the subregion number containing the point. If only 1 subregion has been defined, it is not necessary to specify this variable. ie The format of line 1 above becomes IDB,XBEN,YBEN,ZBEN
NAMEB	is a name for the benchmark location for user convenience. It consists of up to 16 characters.

Notes :

1 The two lines above are repeated for each benchmark point

2 The input of benchmarks is terminated by the following line

-99,0,0,0,0

This line is not followed by a NAMEB line. It is followed by the next keyword

3 There is no fixed limit on the number of benchmarks which may be defined. The only limitations are imposed by disk space, volume of printout and CPU time for benchmark calculations.

4. No facilities are provided for the automatic generation of benchmark locations for BEAP-DD, since this promotes the use of more benchmarks than are often required.

Keyword **END**

Purpose : **Define an end of file marker**

Format :

The keyword END is the only line required

Variables:

none

Notes :

1 This keyword can be used to define an end file marker even there is more data after this keyword. This can be useful sometimes when comparative runs are being carried out. It is not essential to use this keyword at all, but it is also to ensure that data format above the keyword is correct.

Keyword **HEADING**

Purpose : To define a user heading for a run

Format : USERHED

Variables:

USERHED Up to 64 characters

Notes :

1 The user is urged to vary this heading before every run. It is very two get two different runs confused if the user description does not differ from one run to the next

Keyword **INTEGR**

Purpose : To change integration accuracies to values different from the default values

Format : TR2,TR3,TR4,TR5,TR6,TR10
TRLB

Variables:

The above variables define the transition values at which different order Gauss quadrature formulae will be applied. The transition is based upon the separation ratio of the element or sub-element to the point in question. TRLB is a factor which is applied to TR2,TR3,...TR10 during the calculation of benchmarks. This is necessary because stress calculations require more Gauss quadrature points than do displacement calculations. A recommended value for TRLB is 0.7.

Notes :

1 It is usually not necessary to use the INTEGR keyword.

2 This keyword controls the accuracy with which the influence coefficient matrix is calculated. The variables TR2 to TR 10, together with SSW6 (see SSW keyword) control the integration accuracy. If it appears that the accuracy of calculation of influence coefficients is too low, then one might, typically, double all the values TR2 to TR10 and perhaps also SSW 6 . Approximate guidelines for these variable (based on user experience) are given below.

Variable	Approximate Minimum	Maximum	Typical	Comment
TR2	3	300	6	About twice TR3
TR3	1.5	50	3	About twice TR4
TR4	1.0	25	1.8	About twice TR5
TR5	.5	10	1.0	About twice TR6
TR6	.2	3	0.5	About twice TR10
TR10	0	1	0.3	
SSW6	0.5	100	2	

Increasing the above values will improve the accuracy of calculation of influence coefficients in general. Note regarding variable TRLB: Stresses and displacements at benchmarks are also calculated using the above transition ratios (TR2 to TR10 and SSW 6). However, the stress coefficients are dependent upon an inverse distance cubed function. This means that more accurate integration is generally required. TRLB is a factor by which this integration can be improved (at the cost of extra computation time). Decreasing TRLB will increase the accuracy of calculation of benchmarks. Values between 0.4 and 1.0 are suggested, with a default of 0.7.

4 Two other variables printed in module B903 (Integration ratio for lump lumping and Integration ratio for element lumping) cannot be changed directly by the user, but are changed indirectly by the switch variable SSW 6.

Keyword **ISSW**

Purpose : To define integer control switches

Format :

ISW,IVAL
0,0

Variables:

ISW is the number of the switch to be changed
IVAL is the new value for that switch

Notes :

1 Use a line containing 0,0 to end input for this keyword

2 Twenty integer switches have been allocated for future use in the system. Most of these have not yet been used and are thus set to zero by default. Some of the switches are for source code development purposes, and are thus of no direct concern to the everyday user. Switches which may be of use or interest to the user are as follows:

ISSW 1-4 Not used

ISSW 5 Produce debug information. This should not normally be used since it is primarily for program development and could substantially slow down the execution speed of programs.

ISSW 6 Loop Counter. The default is 5 meaning that screen messages for reading in nodes are displayed for every 5th node that is processed. Alternative values range between 1 and 50 and may be set by the user.

ISSW 7 Used internally. Do not use.

ISSW 8 Menu control 1 = on, 0 = off. If menu control has been set as off, then the user would not be asked for the name of each input datafile name. Instead, the default names, D201,D202 etc would be used.

ISSW 9-10 Used internally. Do not use.

ISSW 11 Iteration counter to keep track of the total number of iterations performed since the data base was initialized.

ISSW 12 Used internally. Do not use.

ISSW 13 Printout verbosity. At present, 0 = full, whilst 1 will result in the suppression of page feeds, thereby producing a more compact form of printout.

ISSW 14	Switch 14 records the number of pages which have been printed (ie sent to the list device/channel). The page numbers are reset to zero when a new project is initialized (M101). thereafter it keeps incrementing by 1 until reset again using M101 or using the ISSW keyword.
ISSW 15	Switch 15 records the number of times that D506 has been run.
ISSW 16-19	Not used
ISSW 20	Number of decimal places for plotted numbers.

Keyword **ITERAT**

Purpose : Change default iteration parameters.

Format : NITER,OMEGA,DEL,NITSUB,ACCEL

Variables:

NITER	Maximum number of iterations permitted
OMEGA	Relaxation factor (applies to sub-iterations)
DEL	The required convergence ratio. This is the ratio of the change is displacement for the last and first iterations. Values of DEL range from 0.005 to 0.0005
NITSUB	Number of sub-iterations for each element.
ACCEL	Acceleration factor similar to OMEGA above, except it applies from one iteration to the next, instead of from one sub-iteration to the next as is the case for OMEGA

Notes :

1 It is normally NOT necessary to use this keyword

2 If convergence problems are encountered, then try reducing the value of OMEGA to between 0.4 and 0.8 whilst setting NITSUB = 1 and ACCEL = 1. Convergence should normally occur in from 10 to 30 iterations.

Keyword **PRIM0**

Purpose : To define constant primitive stresses

Format : PXX,PXY,PXZ
PYX,PYY,PYZ
PZX,PZY,PZZ

Variables:

PXX	Constant primitive stress in x-direction
PXY	Constant primitive x-y shear stress = PYX
PXZ	Constant primitive x-z shear stress = PZX
PYY	Constant primitive stress in y-direction
PYZ	Constant primitive y-z shear stress = PZY
PZZ	Constant primitive stress in z-direction

Notes :

1 Positive stresses are compressive

2 The format may use three lines as shown above, or may be compressed into one line as

PXX,PXY,PXZ,PYX,PYY,PYZ,PZX,PZY,PZZ

3 Usually, the shear components are zero

4 It is imperative that the shear components are defined correctly if these are non-zero. It is imperative that the following equalities hold

PXY = PYX
PXZ = PZX
PYZ = PZY

5 These primitive stresses will automatically be applied to all elements whose boundary condition code (KOD) is equal to 1 denoting a mined element. They will not be applied to interface elements.

6 These stresses may be added to those defined by the PRIM1 keyword, by simply defining both keywords in the same data file

Keyword **PRIM1**

Purpose : To define depth dependent primitive stresses

Format : DXX,DXY,DXZ
DYX,DYY,DYZ
DZX,DZY,DZZ

Variables:

The variables DXX,...DZZ have units (stress/length) and are defined in similar way to PXX,....PZZ for constant primitive stresses defined for the PRIM0 keyword. The depth is defined here as being equal to the z-coordinate axis. For example, the actual primitive stress pxx in the x-direction is calculated from DXX as

$$pxx = DXX * z$$

where

z is the depth of the point in question

Notes :

1 Compressive stresses are positive

2 Often, the global z-axis for definition of nodal coordinates will be positive upwards, with the origin at surface. When this happens, all depths (z-coordinates) will be negative and DXX,...DZZ must also be negative to give compressive primitive stresses

3 The total primitive stresses are the sum of those defined by the keywords PRIM1 and PRIM0 if both have defined. ie:

$$pxx = DXX * z + PXX$$

4 Be sure to read the notes for the PRIM0 keyword

5 For example, if the horizontal stress in the X-direction is twice the vertical stress and the horizontal stress in the Y-direction is 1.5 times vertical, then for a vertical stress determined from overburden loading, with units of meters and the Z-axis pointing upwards with origin at earth's surface, typical values for PRIM1 would be :

-0.0540, 0.0000, 0.0000
0.0000,-0.0405, 0.0000
0.0000, 0.0000,-0.0270

Keyword **SPTRAC**

Purpose : To apply special load conditions to elements over and above the primitive stresses

Format :

LB,LE
NB,NE,TX,TY,TZ

Variables:

LB and LE	are as defined above (or use 0,0) to end this section of input
NB	First function node to apply tractions
NE	Last function node to apply tractions
TX	Applied traction in X-direction (shear component)
TY	Applied traction in Y-direction (shear component)
TZ	Applied traction in Z-direction (Normal component)

Notes :

1 Examples of the use of this keyword are for simulating artificial support or for simulating the internal loads generated by caved or backfill material.

2 It is also possible to apply separate tractions to each of the function nodes within a single element. This provides an approximate means of applying point loading

3 The line NB,NE,TX,TY,TZ is continuously read in until NE=9, whereupon the next line containing LB,LE is read.

For example

LB,LE	
1,3,TX,TY,TZ	Apply TX,TY,TZ to nodes 1 to 3
4,7,0,0,TZ	Apply 0 , 0,TZ to nodes 4 to 6
8,25,tx,ty,tz	Apply tx,ty,tz to nodes 8 to 25
LE,LE	Start with next group of elements

Keyword **SSW**

Purpose : To define real control switches

Format :

ISW,RVAL
0,0

Variables:

ISW is the number of the switch to be changed
RVAL is the new value (real) for that switch

Notes :

1 Use a line containing 0,0 to end input for this keyword

2 Valid switches are as follows

1 Current module number being executed

2 Disk I/O monitor. Disk I/O monitor gives an indication of the amount of disk I/O to the random access files used by the program. Because this can be quite large, considerable time (and wear and tear) savings can be gained by putting the LUMSTOR.DD files into virtual disk. SSW2 is automatically set to zero for a new run.

3 Allowable curvature before for lumps before warnings are issued. This value may be changed by the user, and will be used to trigger the warning message regarding curved elements. For displacement discontinuity elements, useful values lie between 0.99 and 1.01.

4 Allowable aspect ratio for lumps before warnings are issued. (Similar to SSW 3 above). This is the ratio of the longest to the shortest side of an element. Warning messages are issued if this ratio is exceeded. This could be set to about 1.01 here if required.

5 Counter for numerical integration. This counter is for interest only, and gives an indication of the number of sample/integration points used to set up the influence coefficient matrix.

6 Scale factor for lumping transition values. This is an important switch and can be used to change the amount of lumping that will be used in a run (and also the amount of disk space). Bigger value result in less lumping. Less lumping results in improved accuracy at the expense of increased disk space and execution time.

Typical values range between 0.8 and 4. Examples are given in Ref. 1. However, there is no hard and fast rule regarding the use of this lumping control. If its value is too low, and lumping is used extensively (producing many 1's in the lumping map of module (D501)).

7	Internal to program - do not use.
8	Internal to program - do not use.
9	Internal to program - do not use.
10	Internal to program - do not use.
11	x-offset for coordinate (Internal to program)
12	y-offset for coordinate (Internal to program)
13	z-offset for coordinate (Internal to program)
14	Internal to program - do not use.
15	Internal to program - do not use.
16	Internal to program - do not use.
17	Internal to program - do not use.
18	Scale factor applied to displacement vectors before plotting
19	Scale factor applied to stress crosses before plotting (this is merely for compatibility with QUAD, since no modules of BEAP-DD will produce this type plot)
20	Scale factors for numbers before plotting. For example, if it is required to plot extension strains as micro-strains, instead of actual strain values, then a scale factor of 1.E6 or 1000000 would be used here.

Keyword **SUBS**

Purpose : To define elastic properties, symmetry code and density for each subregion

Format :

NSUB
MR,E,P,N,DENS

Variables:

NSUB number of subregions (Usually 1)

MR subregion number whose properties are being defined

E Young's modulus for the subregion

P Poisson's ratio for the subregion

N Symmetry code for the subregion

- 1 for no symmetry
- 2 for X symmetry
- 3 for Y symmetry
- 4 for z symmetry
- 5 for XY symmetry
- 6 for XZ symmetry
- 7 for YZ symmetry
- 8 for XYZ symmetry

Notes :

1 Default planes are defined at X=0, Y=0 and Z=0. Some or all of these planes can be activated using N as defined above. If different planes are required, then one should also use the SYM keyword

2 Nodes may be specified to lie on a plane of symmetry, but elements should never be defined on a plane of symmetry.

3 BEAP-DD does not yet support symmetry. Therefore, the value of SUBS must be set at 1 here.

An example of the D207 data file for two intersecting planes problem is given below -

```
HEADING
BIGDD FAULT RUN 3 TO TEST NO FAILURE
SUBS
1
1,100,.3,1,0
PRIMO
+10,0,0,0,+10,0,0,0,+5
ITERATIONS
30,1.2,0.001,4,1.20
BENCH
1 1.8 50 4.7
POINT 1
2 5.4 50 4.10
POINT 2
3 9.00 50 3.50
POINT 3
4 12.60 50 2.90
POINT 4
5 16.20 50 2.30
LAST POINT
-99 0 0 0
END
```

3.6 D503 -- DEFINE BACKGROUND LAYOUT

The purpose of this data file is to set all the element codes to their pre-mining states. For example, fault elements will be set as "Un-failed faults", while unmined seam or reef areas will be defined as seam elements (code 2) if it is to be modelled as deformable, or as dead elements (code 6) if unmined seam deformations are negligible. The format of this data file is identical to that for D505 described in the next section.

3.7 D505 -- CHANGE MINE LAYOUT FOR A MINING STEP

This data file contains instructions for change element codes from one mining step to the next. Usually, the layout will be adjusted by using polygons which have defined the layout. However, blocks of elements may also be adjusted by using a series of lumps or a specific group. The format for this data file is described below.

ISTEP,IPOL,JGR,JFROM,JTO,JKOD

where

ISTEP	is the step number in which changes are to be made.
IPOL	is the polygon number (use O if no polygon is to be used).
JGR	is the group number to which changes are to be made (use O if no specific group is to be used).
JFROM	Start lump number of a sequence of lumps which are to be used.
JTO	Stop lump number for sequence of lumps which are to be used. Set JFROM=JTO=0 if all lumps are to be used (subject to the restrictions imposed by IPOL and JGR above).
JKOD	is the "DD code" or element code to be used. For example, code 1 denotes a mined element.

For a single seam problem, it will usually not be necessary to specify JGR, JFROM and JTO as the use of polygons will be sufficient to change mining layouts.

However, when faults on folded seams or multi-seams are used, then it will usually be necessary to use JFROM, JTO and / or JGR to ensure that the correct lumps are checked against specified polygons.

An example of data file D505 is given below:

```
1,5,0,0,0,1
1,6,7,0,0,1
1,7,0,0,0,1
1,8,0,0,0,1
2,1,0,0,0,1
2,2,0,0,0,1
2,3,0,0,0,1
2,4,0,0,0,1
```

This data file will convert all elements inside polygons 5 to 8 to "mined" for step number 1 and all elements inside polygons 1 to 4 to "mined" for step number 2.

3.8 D515 READ IN A MINE LAYOUT AS AN ARRAY

This data file contains instructions for change element codes from one mining step to the next. If the layout is not be adjusted by using polygons, then the alternative is actually to read in the element codes directly. The format for this data file is described below.

NROW,NCOL
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
KKKKKKKKKKKKKKKKK
ISTEP,0,JGR,JFROM,JTO,JKOD

where

NROW	is the number of rows to follow describing the mine layout. ie The number of lines of type 'KKKKK' to follow.
NCOL	is the number of columns per row describing the mine layout. ie The number of columns for each line of type 'KKKK' which follow.
K	is an ASCII code describing the element code or status. Valid codes are as described under the DD codes below.
ISTEP	is the step number in which changes are to be made.
JGR	is the group number to which changes are to be made (use O if no specific group is to be used).
JFROM	Start lump number of a sequence of lumps which are to be used.
JTO	Stop lump number for sequence of lumps which are to be used. Set JFROM=JTO=0 if all lumps are to be used (subject to the restrictions imposed by IPOL and JGR above).
JKOD	is not used here but is left for compatibility with D505. (Use 1)

For a single seam problem, it will usually not be necessary to specify JGR, JFROM and JTO

However, when faults on folded seams or multi-seams are used, then it will usually be necessary to use JFROM, JTO and/or JGR to ensure that the correct lumps are checked against specified array.

An example of data file D515 is given below:

```
20,60  NO OF ROWS AND COLS
++BBBB+B+BB+BB+BBBBBBBBBBBB+BBBBBBB+BBBBBBBBB+BBBBBBBB++++++
++BBBB+BBBB+BB+BBBBBBBBBBBB+BBBBBBB+BBBBBBBBB+BBBBBBBB++++BB
+BBBBB+BBBBBBB+BBBBBBBBBBB+BBB+BBBB+BBBBBBBBB+BBB+BBBBB++BBB
+BBBBB+BBBBBBB+BBBBBBBBBBB+BBB+BBBB+BB+BBB+BB+BBB+BBBBB++BBB
BBBBBB+BBBBBBB+BBBBBBBBBB++BB+BBB+++BBBBB+BB++BBB+BBBBB++BBB
BBBBBB+BBBBBBB+BBBBBBBBBB++BB+BBB+++BBBBB+BB++BBB+BBBBB++BBB
++++++++++++++++++++BB++++++++++++++++++++++++++++++++++++BBB
++++BBBBBBBB+BBBBBBBBB++++++++++++B+BBBBBB++++BBBBBBBBB++BBB
++++BBBBBBBB++BBBBBBBBBBBBBBBBBB+++BBBBB++++++++++++++++BBB
++BBBBBBBBBBB++BBBBBBBBB+BBBBBBBB++BBBBBB+BBBBBBBBBBB++BBB
+++BBB+BBBBBBB+++++++++B++++BB+BBB+BB+B+BB++++++++++++B++BBB
+++BBB+BBBBBBB++BBBBBBBBB++BBBBBBB+BBBBBBB++++BBBBBBBB+++BBB
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++BBBBBBBB++BBB
++BBBBBBBBBBBBBBBBBB++++++++++++++++++++++++++BBBBBBBB++BBB
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++BB
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++BBB
++++++++++++++++++++++++++++++++++++++++++++++BBBBBBBB++BBB
BBBBBBBBBBBBBBBBBBBB++++++++++++++++++++++++++BBBBBBBB++BBB
1,0,0, 1,48,1
```

This layout defines a unmined and backfill element codes for lumps 1 to 48 for mining step number 1.

This data file has been used for some of the plots for the main report.

3.9 CODES FOR ELEMENT TYPE DEFINITION

Displacement discontinuity elements represent different types of material and/or stages of mining. Each element is assigned a code (DD code) describing whether it is solid, unmined, mined out, backfilled etc. The various element codes are summarized in the table below and are described in more detail in the following pages.

TYPE	INTEGER CODE	ASCII CODE	ALTERN-ATIVES	DESCRIPTION
IMIN	1	M	ITOT	Mined out
ISOL	2	+	-	Unmined (Deformable solid)
IBAC	3	B	-	Mined and backfilled
IFLT	4	F	ITEN,ISHR	Fault (pre failure)
INLS	5	N	INLD,INLR	Non-linear solid undergoing down-loading
IDED	6	.	-	Dead element
ITOT	7	C	IMIN	Total closure
ISHR	8	S	ITEN	Shear failed fault
ITEN	9	T	ISHR	Tensile failed fault
INLD	10	D	INLR	Non-linear solid undergoing down-loading
INLR	11	R		Non-linear sold at residual strength
IPAR	12	P		Partially mined element
IZET	13	Z		Zero traction element

Note :

Elements currently available in Version 1.0 of BEAP-DD are
1, 2, 3, 4, 6, 7, 8, 9 and 13.

Others have not yet been included.

Further details about the use of these codes are presented individually below.

DD Code	**IMIN** -- Mined Element
Purpose :	Normal mined element.
Features :	Tractions applied automatically, based upon primitive stresses. Additional tractions can also be applied (eg caved material etc).
Integer Code :	1
ASCII Code :	M

DD Code	**ITOT** -- Totally closed element
Purpose :	To model total closure.
Features :	Total closure with no shear resistance.
Integer Code :	7
ASCII Code :	C

Notes :
1. Only code IMIN can become totally closed.

DD Code	**ISOL** -- Solid Element
Purpose :	Normal unmined element.
Features :	Stiffness based upon linear springs for normal and shear displacement discontinuity.
Integer Code :	2
ASCII Code :	+

DD Code **IBAC -- Backfill Element**

Purpose : To model any type of backfill or support.

Features : Can be used to model backfill of order 0, 1, 2 or 3. Can also be used to apply special tractions.

Integer Code 3

ASCII Code B

Notes:

1. A backfill element will generally not have a total closure restriction. The backfill stiffness should be sufficient to ensure that total closure never occurs in actual modelling.

2. The five backfill parameters are as follows :

   FILL1 Normal support applied as an active stress independently of closure. This could represent, for example, the loading from hydraulic prop type supports. Or they could be used if a given backfill load is desired, independently of closures.

   FILL2 The percent strain required at an element before any closure of the backfill (and hence backfill loading) commences.

   FILL3 The order of the polynomial for the fill curve. ie 2 = quadratic, 3 = cubic. In other words a stiffer fill could be modelled with a higher order fill curve.

   Fill4 The multiplier to be applied to the passive backfill load. For example, for

3. The backfill routine starts by calculating the 'stope strain as a percentage'

   stope strain = 100 * closure / (stope height)

   Then this is compared with FILL2 to see whether any closure of the fill has taken place. If not, then only the active stress loading, as specified by FILL1 is applied. If closure of the fill has taken place, then a fill strain as a percentage is given by

   fill strain = stope strain - FILL2

   The fill stress is then given by

   Fill Stress = FILL4 * (fill strain) ** FILL3 + FILL1

   Finally, the shear stress components are obtained in a similar way to the normal component of stress, except that they are multiplied by FILL5.

4. For example, assume the following values for an element:

   FILL1 0.5 MPa
   FILL2 5%

FILL3	2
FILL4	0.02
FILL5	0.025

Then for a closure of 30 %, and a ride component of 10%, the fill strain is 30% - 5 % = 25% (approximately), and the normal stress is

0.5 MPa + 0.02 * (25)**2 = 13.0 Mpa

The shear stresses generated would be

10% * FILL4*FILL5*(FILSTR**FILL3)

10% * 0.02 * 0.025 * (25)**2 = 3.125 Mpa

5. It is easy to produce a fill curve to check fill input parameters after a run has been completed. Merely plot the stresses vs closures for a number of backfill elements.

DD Code	**IFLT** -- Fault Element
Purpose :	To model a fault.
Features :	Mohr-Coulomb failure
Integer Code :	4
ASCII Code :	F

Notes :

1. This code only applies until such time as the fault fails, whereupon the status automatically changes.

2. If the fault fails in shear, then the element code is changed to ISHR. The failure criterion is a standard Mohr Failure criterion depending upon the cohesion, angle of friction (see D203) and normal stresses acting on the element.

3. If the fault fails in tension, then the element code is changed to ITEN. The tensile strength is also defined in D203.

4. If a fault element which has opened up closes again, then the element is then treated as being frictionless, but the element code stays set as ITEN, showing that tensile failure took place at some previous mining step.

DD Code	**ISHR** -- Shear failed fault element
Purpose :	To model shear failure on a fault (See IFLT)
Integer Code :	8
ASCII Code :	S

DD Code	**ITEN** -- Tensile failed fault element
Purpose :	To model tensile failure of a fault element (See IFLT).
Integer Code :	9
ASCII Code :	T

DD Code	**IZET** -- Zero traction element
Purpose :	Modelling earth's surface.
Features :	Zero traction element, irrespective of location or primitive stresses.
Integer Code :	13
ASCII Code :	Z

DD Code	**IDED** -- Dead Element
Purpose :	To model a dead element (ie there will be nothing at that location).
Features :	Used to reduce problem size.
Integer Code :	6
ASCII Code :	.

Note :

1. This element type may also be used to model an incompressible seam.

3.10 USE OF DD ELEMENTS TO MODEL THE EARTH'S SURFACE

Although DD elements are not ideally suited to modelling the earth's surface, they can, however, be used very easily. Usually, when the earth's surface is to be modelled, it will be defined as a plane located at Z=0. Various possible configurations of lumps which could be used to define the earth's surface are shown in Figure 3.10.1 . A number of points should be noted from this figure.

- A straight line outcrop can be accommodated.
- Use of bigger lumps further from areas of under-mining will often be more efficient than many smaller lumps.
- Lumps do not need to be aligned with coordinate axes.
- There is no hard and fast rule to determine how far from an area of undermining the earth's surface should be modelled. For any given layout, however, closures and rides in the "surface" elements should be near zero, as should the stresses at test benchmarks located at Z=0, a few element widths away from the edge of the surface grid/layout.
- If the BEAP-M program is available, then it would be better suited to modelling such problems, since boundary elements would more effectively used for modelling the earth's surface.

When underground elements are close to the earth's surface, the "surface" elements should be aligned parallel to the strike of the underground elements wherever possible, and element sizes should match. Element sizes should always be chosen so that no elements are within one element width of each other (unless they are equally sized and intersecting along their edges).

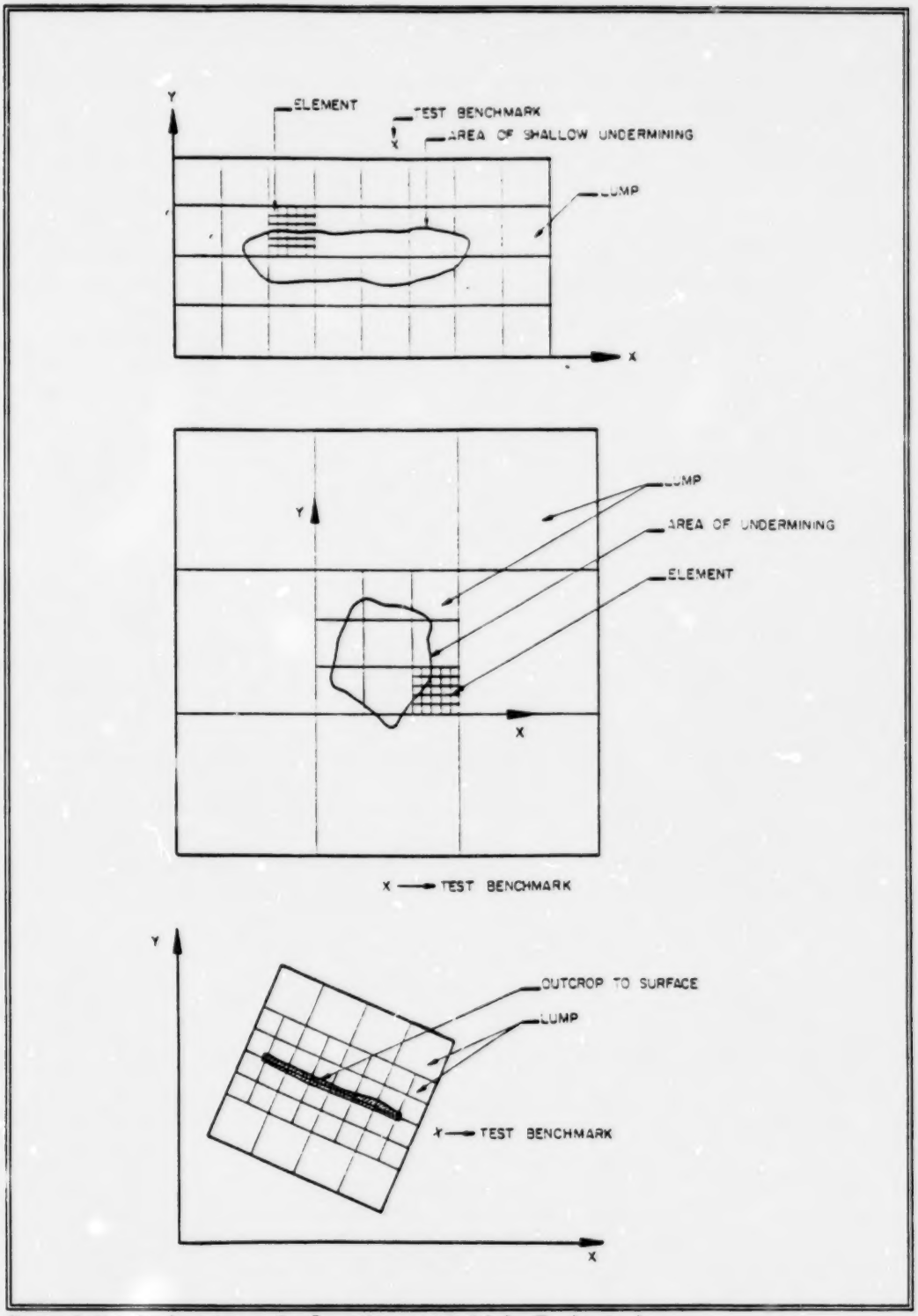

Figure 3.10.1: Various Lump Configurations to Model the Earth's Surface

4.0 DESCRIPTION OF PROGRAM MODULES

In the following paragraphs, each program module making up the BEAP system is described. Program modules may be run either via the program menu driver, or via DOS as a command. For example D204 is the fourth choice of menu number 2, or D204 may be run from DOS simply by entering:

D204

at the DOS prompt. It is worth noting here that DOS batch files can easily be built to enable many modules to be run at once. The user responses can also be built into response files which can be activated using the DOS piping capability. Refer to the DOS manual for further details.

When each program is run, the printout is written to a file TEMP.LST. If this printout is required as hard copy, then program control must be returned to DOS, and the file TEMP.LST sent to the printer using a DOS PRINT or COPY command. If programs are being run under batch control, then the PRINT commands can be built into the batch file. If the next program starts before the printing of the previous run is complete, then some of the printout can be lost. To avoid this situation, the DOS COPY command is sometimes preferable, since it waits until the printout is complete before proceeding.

BEAP Menu driver for BEAP and BEAP-M

This program enables BEAP to be run under menu control. Before this program will start, the files AUTOEXEC.BAT, CONFIG.SYS must have been previously adjusted as described under the installation procedures. In addition, there should be at least 400k free system memory available and at least 2 Megabytes of free disk space. In addition, the file MESAGE.BE must have been located in the subdirectory BEAP on the C drive (ie C:\BEAP\MESAGE.BE must exist).

M101 Initialize an entire new database (BEAP and BEAP-DD)

M101 will initialize all database files for new run. **NB** This includes the database files for BEAP and BEAP-M if these programs are being run at the same time. It will create empty files (not necessarily with 0 bytes, however, since certain direct access files must contain an initial header record). A list of the files created is as follows:

List of files for BEAP, BEAP-M and BEAP-DD

BEMCO.	BE	Influence coefficients (BE)
BEMMESH.	BE	Plot file of BE mesh
BEMMESHE.	BE	Plot file of BE element numbers
BEMMESHN.	BE	Plot file of BE node locations (as X)
BEMPOL.	BE	Polygon master file
BENSTOR.	BE	Benchmark file (Binary)
BOTH.	BE	Basic information (internal to program)
BVECT.	BE	Right hand side vector
LIND.	BE	Lumping maps for (BE)
LUMPS.	BE	Geometric information for BE elements
LUMSTOR.	BE	Detailed information for BE elements
LUMSTOR2.	BE	Second file for BE elements
SELFCO.	BE	Element self-coefficients (BE)
VARCOM.	BE	Basic information (to supplement BOTH.BE)
XN.	BE	Nodal coordinates
TEMP.	LST	printout file for results

Only used by BEAP-M

COEF.	BD	Influence coefficients (BE-DD)
LIND.	BD	Lumping maps (BE-DD)
BVECT.	DB	Right hand side vector
COEF.	DB	Influence coefficients (BE)
LIND.	DB	Lumping maps (BE-DD)

Used by BEAP-DD (and hence BEAP-M also)

BEMCO.	DD	Influence coefficients (BE)
BEMMESH.	DD	Plot file for DD elements
BEMMESHE.	DD	Plot file of DD element numbers
BEMMESHN.	DD	Plot file of DD node locations (as X)
BEMSTEP.	DD	Store information for multi step runs
BVECT.	DD	Right hand side vector
LIND.	DD	Lumping maps (BE-DD)
LUMPS.	DD	Geometric information for DD lumps
LUMSTOR.	DD	Result information for DD elements
LUMSTOR2.	DD	More information for DD elements
LUMSTORG.	DD	Geometric information for DD elements
POLPLOT.	DD	Plot file for polygons
SELFCO.	DD	Self effects for DD elements
SHEET.	DD	File to store spreadsheet information
STORSTEP.	DD	Store information for multi step runs

D102 Initialize polygon master file

This module will set the number of polygons in the polygon master file equal to zero. ie Previous polygons entered into the system will be erased. If M101 is to be run, the polygon master file will also be initialized. Therefore, if polygons are to be saved, the file should backed up under DOS control.

D103 Initialize closures, rides and tractions

This is a useful module which initializes only the closures, rides and tractions for each element. It can be run if all geometric information has been read in correctly, but there was, for an error or adjustment in D207, or if a new run is to be started with the same geometry, but only changes to the D207 file. If the user is uncertain as to it's use, then it will be safer to run M101 and start a completely new run.

D201 Read in node and DD lump geometry (file D201)

This module reads the first of the user data input files (D201). The user is prompted for the name of his file. Thereafter, various information is displayed to screen to show progress as the file is read. If a data input error is encountered, the program may crash. If this is the case, then correct the data error and run M101 and D201 again. If the data is read in successfully, then proceed with D202, D301 or D204 according to your requirements.

D202 Read in lump grouping (file D202)

This module reads in the grouping of lumps into separate grids as well as the initial element codes, pen collars for lumps and excavation numbers for easier processing of lumps elsewhere.

Various geometric checks are carried on the lump geometry. If any warnings are produced either to the screen or the print file TEMP.LST, then these should be checked before proceeding further.

D203 Read in lump material properties (file D203)

D203 reads in the D203 datafile which contains material properties for each lump.

D204 Print selected displacement discontinuity lumps (Summary)

This enables the user to obtain a printout of selected nodal coordinates and summary information about the lump geometry (usually to facilitate checking the input data). Printout is directed to file TEMP.LST.

D205 Print selected DD lumps (detailed)

This is similar to D204, except that detailed information about each lump is printed. ie Coordinates and area of each element within the selected lumps are printed. This printout, also directed to print file TEMP.LST, can be very lengthy.

D206 Print out summary of polygons

Heading, coordinate system code and number of coordinate pairs for each selected polygon are printed to print file TEMP.LST.

D207 Read in remaining input (file D207)

This module, which is identical in function to M401 of BEAP, will read in data file D207 containing the 'general input' for the run, including heading, benchmarks, primitive stresses etc.

This module is also used to set the default plot file preparation parameters (ie stress / Vector scale factors, number of decimal places, scale factor for numbers written out and character size) See the SSW keyword under data file D207 for further details.

D208 Print header information

This will print out a header page containing a summary of most of the information read in under D208. Printout is directed to the print file TEMP.LST.

D209 Print lump summary information (one line per lump)

This is also similar to D204, except that slightly different information is printed.

D301 Generate a grid of lumps

This is used to convert a single lump, as defined by a data file of type D201 into an array of lumps. For example, if it is desired to construct two 'grids' of lumps each consisting of 10 X 10 lumps (as for the shallow reef example), then the datafile D201 would be prepared so as to contain only two lumps (lump number 1 and lump number 101). Using this module, lump number 1 could be expanded into a 10 X 10 grid of lumps (number from 1 to 100). Next it would be used to expand lump number 101 into a second grid of 10 X 10 lumps numbered from 101 to 200. Once this has been run, D204 could be run again to produce a listing of newly generated nodes and lumps. The printout (file TEMP.LST) can itself be edited to produce a new file D201 for future runs or for backup.

When running D301, you are prompted interactively for the number of rows and columns required. Also, you are prompted as to whether each lump should or should not be generated. This enables an incomplete or partial grid of lumps to be generated, saving computer processing time in some cases in which scattered mining might exist.

D401 Prepare a 3-D view of mesh/elements/nodes

This is an extremely useful debugging tool. The user can obtain a 3-D isometric view, and X-Y plan plot, or X-Z, Y-Z section plots (ie looking north, south, east or west). The user may also select which lumps are to be plotted. Plot information is routed to the files BEMMESH.DD, BEMMESHE.DD and BEMMESHN.DD.

D402 Prepare polygons for plotting

This enables a plot file (POLPLOT.DD) to be prepared for user selected polygons from the polygon master file. These may then be plotted with other plot files, such as stress or closure plots.

D502 Calculate lumping maps (DD-DD)

This module calculates the lumping maps for displacement discontinuity lumps (DD-DD). The effect of each lump on each other lump is assessed and the type of lumping or element-element interaction required is assessed. The following codes are displayed to the screen and to the print file TEMP.LST.

As with BEAP, a map is set up indicating how each lump is related to each other lump. These values are stored in the array LINRE for each lump. The table below gives details concerning different types of coefficient.

LINRE	NCOLR	NCOLE	File	IDD	Name	
1	1	1	Not stored	1	Lump-lump	DD coefficients
2	1	1	BEMMDD.BE	2	Lump-lump	BE coefficients
3	1	25	BEMMDD.BE	2	Lump Elt	Be coefficients
4	25	1	BEMMDD.BE	2	Elt lump	BE coefficients
5	-	-	-	0	Not used yet	
6	-	-	-	0	Not used yet	
7	25	25	BEMDD.BE	2	Elt-Elt	BE coefficients
8	25	25	Not stored	1	Elt-Elt	DD coefficients
9	25	25	Not stored	3	Self coefficients	

In this table, NCOLR is the number of receiver "nodes" and NCOLE the number of emitter "nodes". When lumps with the same size and orientation are located in the same plane, it is possible to make use of a standard set of influence coefficients as was the case for program MBEM. Under these conditions, IDD is set to 1. If the emitter and receiver elements both lie in the same lump, then IDD is set equal to 3 and LINRE = 9. However, when two lumps have different size, orientations or are not co-planar, then the influence coefficients between these two lumps have to be evaluated and stored in disk files. Under these conditions, IDD is set to 1 and LINRE has a value of 2,3,4 or 7 depending upon the relative sizes and separations of the lumps.

Obviously, those coefficients which require to be stored on disk will occupy disk space and will also require time to be calculated. Both this space and time are very sensitive to the particular problem being solved. For large problems, the space and time can be prohibitive. The lumping maps and requirements are calculated in program module D501. Output from this module must be studied by the user whether sufficient disk space is available to store the coefficients. If not, the problem must be simplified, or the accuracy of the integration reduced.

When IDD = 1, influence coefficients may be calculated in closed form or using numerical integration. When IDD = 2, however, the coefficients must be evaluated using numerical integration.

The total size which will be required by the file BEMCO.DD by D502 is calculated and reported in TEMP.LST. The user should inspect the printout and check whether sufficient disk space is available for the remainder of the run. If the space required by these files is not available, then the amount of space may be reduced (at the expense of reduced accuracy) by decreasing the value of real switch number 6

```
SSW
6,1.6
0,0
```

For example, the above segment of data file D207, if read in via D207 will set the lumping transition ratio to 1.6. Values between 0.8 and 4 will usually be okay. If too much lumping is being used, poor accuracy will result. The correct choice for this ratio is best found by experience and the accuracy requirements of the user.

The lower the value of this ratio, the more 1's will be produced in the lumping map and the less will be the storage requirements. (see reference 9 for further information)

D502 Calculate influence coefficients (DD-DD)

This module calculates the effect or influence of each displacement discontinuity element on every other displacement discontinuity element. The level of detail or accuracy which is used for each set of coefficients depends upon the integration parameters as set by the INTEGR keyword. If this keyword is not specified, then the default values will be used, which should be acceptable for most runs. See further details under the INTEGR keyword description.

D503 Initialize layout codes prior to first step (D503)

This module will read in file type D503 to initialize the background 'mining' codes for each lump and the elements of each lump prior to the first mining step. For example, all elements may be set as

solid/unmined or fault elements may be so defined. The program will display a summary of how many elements are changed as this module is run.

D504 Increment mining step number

This merely increments the mining step by one and displays the new mining step number on the screen (and print file TEMP.LST). If it is being run for the first time, then the mining step number will be set to 1. (ie the mining number is set equal to zero by M101). If one is trying to read in a mine layout using D505 or D515, and nothing appears to be changing, then it is quite probable that there is a mining step number mismatch. ie If the layout is defined for step number 1 and the current mining step number is 2, then no layout changes could be obtained using D505 or D515.

D505 Update layout codes for current step (D505)

This module processes file type D505. The program will display a summary of how many elements are changed as this module is run. Element codes may be set either by polygon, by group, by excavation number or by lump number or a combination of these. See separate documentation for the preparation of the appropriate data file.

D506 Apply tractions due to primitive stresses

D506 will apply the primitive tractions (as defined by the PRIM0 and PRIM1 keywords to all elements whose boundary condition is 1 representative of a 'mined' surface. If there is uncertainty as to what values have been applied, then the user can run D205 to check the actual traction values at each element.

D507 Solve equations (DD)

The displacement discontinuity element equations are solved in an iterative manner. If the problem has been correctly defined, then the closures and rides should slowly converge towards the 'correct' solution. The rate of convergence is controlled by the ITERAT keyword. If this keyword is not present in the D207 data file, then the default values will be used. If the error (actually an estimate of the stress imbalances at each element from one iteration to the next) displayed at each iteration is not converging satisfactorily, then the element mesh or configuration may be in error and the geometry and material properties should be checked.

This module will often take several hours to run, and is thus usually best run as part of a batch run.

D507 will read the array coefficients of the equations from the files BEMCO.DD. The updated closures, rides in element stresses are written into files LUMSTOR.DD and LUMSTOR2.DD. In addition, other information is updated in files BOTH.BE and LUMPS.DD.

D508 Calculate benchmarks stresses and displacements (DD)

D508 will evaluate the stresses and displacements at each specified benchmark point (as set using the BENCH keyword). Results are written to the printout file TEMP.LST and are also written the file BENCH.BE (binary format) for subsequent processing by BEMPOST or D606.

It is suggested that benchmarks do not be located closer to an element than one and a half times the width of an element. If benchmarks are too close to an element, then this is usually characterized by artificially high tensile stresses being calculated.

The accuracy for calculations for benchmarks which are close to an element are particularly dependent upon the specified integration accuracy (see INTEGR keyword).

Benchmark printouts contain four main sections for each benchmark point. These are:

- Location and displacements
- Stresses
- Principal Stresses
- Accuracy/Integration information

Each section is described below.

(i) Location:

IBEN	IDB	NAMEB
XB	YB	ZB
UB	VB	WB

where	IBEN	Benchmark number
	IDB	User number for benchmark
	NAMEB	User description for benchmark
	XB,YB,ZB	Location of benchmark
	UB,VB,WB	Displacement components of benchmark

(ii) Stresses:

IXX	IXY	IXZ
	IYY	IYZ
TXX		IZZ
TXY	TYY	
TXZ	TYZ	TZZ

Where IXX ... IZZ are induced stresses. For example, IYY is the induced normal stress in the Y direction, while IXZ is the induced shear stress components in the XZ plane. Remember Total Stresses = Induced stresses + Primitive stresses where primitive stresses are as defined by the PRIM0 and PRIM1 keywords.

TXX, ... TZZ are total stress components similar to IXX ... IZZ above.

(iii) Principal Stresses

PMIN	PINT	PMAX
DMIN	DINT	DMAX
DDMIN	DDINT	DDMAX

Where:

PMIN, PINT, PMAX are minimum, intermediate and maximum principal stresses

DMIN, DINT, DMAX are dip vectors for the orientation of the principal stresses measured from the X-Y plane towards the -ve Z axis (i.e. a lower hemisphere projection).

DDMIN, DDINT, DDMAX Defines the orientation of the principal stresses in the X-Y plane. A value of zero implies that the projection of the principal stress onto the X-Y plane lies along the +ve X-axis. 90 corresponds with the +ve Y-axis, + or - 180 corresponds with the -ve x axis and so on.

(iv) Accuracy/Integration Information

Nearest	at XL, YL
NPTS	MOST =
NUMLUM	NUMELT =
CLOSEST	RATIO =
SUB-REGION	

where Nearest	=	Element closest to the benchmark
XL	=	Local X coordinate of element point closest to benchmark
YL	=	local Y coordinate of element point closest to benchmark
NPTS	=	Number of integration points which were used to evaluate the benchmark
MOST	=	Highest number of integration points used for any one element to evaluate the benchmark.
NUMLUM	=	Number of elements which were not divided into subelements.
NUMELT	=	Number of sub-elements used
CLOSEST	=	Closest distance encountered between elements (Boundary surface) and the benchmark
RATIO	=	Minimum ratio encountered for the ratio of distance squared divided by the area weight of an integration point (See notes below).
SUBREGION	=	Subregion number which contains the benchmark.

Close scrutiny of the above information is usually not necessary for each benchmark. However, it can be useful in deciding whether a particular benchmark is likely to be reliable or not. Benchmark values will tend to be unreliable if (1) the points are too close to the boundary element surface or (ii) if the integration accuracy is too low or (iii) if the points are too close to the edge of an element. The usual symptoms of the above three cases are as follows:

(i) Ratio has a value of less than 9. The minimum principal stress often shows an unrealistically high tensile stress.

(ii) RATIO has a value of less than 9, and MOST has a value of 36 or even 9 or 4. This indicates that although the benchmark was close to an element, automatic refinement of the integration did not take place.

(iii) If CLOSEST is less than one fifth of the square root of the area of the element closest to the benchmark, and XL and/or YL are close to + or - 1, then it is likely that that benchmark is too close to the edge of an element. Other symptoms are unexpectedly large shear stresses or principal stress orientations which change very rapidly from one point to the next.

Thus the information provided in the fourth section of the benchmark printout can assist the user to identify unreliable benchmark points. The user should of course exercise caution when defining the benchmark locations so as to avoid the above uncertainties.

D509 Calculate energy release rates

This module should really only be run after the second mining step has been run. The reason for this is that the energy release rate is calculated as half the normal stresses acting on an element **before** it is mined times the closure **after** it is mined.

The calculated ERR values are stored for each element, and are thereafter available for plotting or print with other BEAP-DD program modules.

Note that if energy release rates are to be calculated, then the information from the previous step must have been stored using D510.

D510 Store element results for future reference

This module will store the calculated closures and stresses at all lumps and elements for the current mining step. The values are written to file STORSTEP.BE and may be retrieved later under program control (for example for printing or plotting). It is usually not necessary to run this module unless, for example, a multi-step is being run overnight, and it is required to print selected results the next day or later.

It is also necessary to run this module if energy release calculations are to be performed using D509.

D511 Store benchmark values for future reference

This is the same as D510, except that it processes the benchmark results. This module has a limit of up to 200 benchmark values. The program will copy the benchmark values from file BENCH.BE to other locations within the file, thus making the size of the bench.be file bigger.

If there are more than 200 benchmarks, which are required to be stored for a single step, then this can be achieved using the DOS COPY command to copy the file BENCH.BE to some backup file name (for example BENCH4.BE)

It is usually not necessary to run this module unless, for example, a multi-step is being run overnight, and it is required to print selected results the next day or later.

Users familiar with the program and with DOS would probably find it easier to save these files under DOS control for future post processing. If the post-processing is carried out before the next mining step is started, then it is also not necessary to run this module.

D515 Read in a mine layout (in array format) for current step

This module provides an alternative means to input a mine layout array from the polygonal method. If this approach is used, then the user will have to prepare an ASCII input data file (see D515 documentation) with the mine layout in a grid format.

D601 Define current step number

This module enables a mining step number to be reduced to some previous value. For example, if five steps had been carried out, and the program was to be re-started from step number 3, then this module would be run, followed by D602

D602 Re-load element results

D602 will reverse the process carried out by D510 for the step number as defined by D601. If D510 was not run for that step number, then it would also not be possible to run D602, since there would be no saved data to restore.

D603 Re-load benchmark results

D603 will reverse the process carried out by D511 for the step number as defined by D601. If D511 was not run for that step number, then it would also not be possible to run D603, since there would be no saved data to restore.

It should also be noted that a maximum of 200 benchmarks could be restored here.

D604 Print selected element results

This module is will print closures and stresses and optionally energy release rates for user selected lumps. Options are available for an array type print for each lump or for tabular printout. Usually the array type printout is preferable. The following menu of options for printout is presented to the user:-

PRINT OPTIONS ARE AS FOLLOWS

1 STANDARD PRINT OPTION ELEMENT BY ELEMENT IN LIST FORMAT
2 STANDARD PRINT OPTION SORTED BY ROW AND COLUMN
3 ARRAY PRINT OPTION -- STRESSES LEFT, CLOSURES RIGHT
4 ARRAY PRINT -- LEFT = NORMAL STRESS,CLOSURE AND LAYOUT
RIGHT = ERR,MAX SHEAR,STRENGTH
5 ARRAY PRINT -- LEFT = NORMAL STRESS,CLOSURE AND LAYOUT
RIGHT = EMPTY

It should be noted that a total printout can be very lengthy, since one page is required for each lump printed. For large problems, graphical display or only display for selected lumps is suggested.

D605 Print current layout as a grid

This module will print the 'mine layout' as an array of ASCII characters, one for each element. The printout from this module would be similar to the array input required by D515.

D606 Print selected benchmark results

D606 enables the user to print user selected benchmarks (after they have been calculated using D508) or after they have been restored using D603. The format of the printout is the same as that produced by D508.

D607 Write out selected results as PC-MINE extract file

The user may select closures, stresses or rides for 'exporting' to the GEMCOM's PC-MINE system. The element coordinates and selected values are written to a binary file in a format which can then be imported into PC-MINE for subsequent processing (for example for contouring). Usually, this module will not need to be run, since a quicker and generally easier approach would be available using D701.

D608 Write out selected results in spreadsheet format

This is a useful module if the closures, rides or stresses along a selected row or column are required to be displayed in graphical format. The graphs of closure and ride for the shallow ride example were produced in this way.

D701 Prepare element data for plotting

This is probably the most useful post-processing utility for BEAP-DD. Certainly, it will produce the best 'looking' plots. A datafile (BEMMESH.DD) is prepared interactively using D701. A menu is supplied to the user as follows:-

SUB MENU FOR ARRAY PLOT PREPARATION

0 RETURN TO MAIN MENU (OR STOP)
1 REWIND PLOT FILE TO START AGAIN
2 DEFINE/EDIT LUMPS TO BE PROCESSED
3 PLOT LUMP OUTLINES
4 PLOT NUMBERS INSIDE EACH ELEMENT
5 PLOT COLOR CODED + SIGNS INSIDE EACH ELEMENT
6 PLOT COLOR CODED ELEMENT OUTLINES
7 DEFINE THE COLOR FILTER
8 DEFINE TRANSFORMATION AND PLOT TYPE

Option 1 is run if the user makes an error and wishes to start over without returning to the main menu. Option 2 MUST be run to select the lumps to be processed. Option 3 will prepare for plotting the lump outline as a square.

Option 4 will prepare a plot file containing the selected value option in number format. Whilst this is convenient for some small examples, the time taken to plot larger plots makes this option unattractive in many instances. In such circumstances, the user would select options 5 or 6 below, which will plot much faster.

For Options 4,5 and 6, the user may select one of the following values for processing/extraction to the plot file.

OPTIONS FOR EXTRACTION OF RESULTS
1 CLOSURES
2 X COMPONENT OF RIDE
3 Y COMPONENT OF RIDE
4 NORMAL STRESSES
5 X COMPONENT OF SHEAR STRESS
6 X-Z SHEAR STRESS
7 ERR (NOT AVAILABLE YET)
8 MINE LAYOUT CODE (1 = mined, 2 = solid etc)

Options 5 and 6 are similar, except that one produces a cross or + sign at the element centre, whilst the other will actually plot the element outline. Option 7 enables a COLOR 'filter' to be defined which will control the collars used by options 4,5 and 6. If COLOR coding is required, then Option 7 must be run BEFORE options 4,5 or 6.

Option 8 enables the user to define a plot as being different from the default x-y plot. For example, if an isometric plot is required (for example to match with a mesh produced using D401), then the transformation and plot type must be defined **before** the other options are run.

D702 Prepare benchmarks for plotting (BEMPOST)

BEMPOST Post processor for benchmarks

This program enables benchmark results to be prepared for plotting. It enables various failure criteria to be tested also, the results of which are contained in the printout.

Plot files which are generated are :

BEMPOST.1	User's first choice
BEMPOST.2	User's second choice (optional)
BEMPOST.D	Displacement vectors
BEMPOST.D3	Third component of displacement (ie the one not included in BEMPOST.D)

These files are in a binary format and require a separate plotting utility before they may be plotted. One example would be GEMCOM'S QUIKPLOT or GEMPLOT utilities. The format of all plot files is described separately, so that other plotting utilities may be modified to read these plot files.

BEMPOST enables coordinates and/or stresses to be transformed according to a user defined view point and origin. In addition, a box may defined to filter those benchmarks which are to be processed. This is useful, in that it enables benchmarks to be processed on a level by level or section by section basis.

BEMPOST will access the current database directly. In particular, it will read the benchmark results from file BENCH.BE. It will obtain subregion material properties from BOTH.BE

Before BEMPOST may be run, the message file BEMPOST.MES must have located in the directory C:\BEAP (ie C:\BEAP\BEMPOST.MES).

D800 Digitizer utilities

The user is presented with another menu, shown below, which at present contains only two options.

DIGITIZING UTILITIES
= = = = = = = = = = = = = = = = = = =

0 STOP
1 SET UP DIGITIZER AND PLAN SCALES
2 DIGITIZE POLYGONS
3 RETURN TO MAIN MENU

Option 1 is used to define the location on the digitizer tablet of the plan or section from which polygons are to be digitized. Option 2 will enable the user to interactively digitize polygons. Up to 178 points may be used for a polygon.

As each polygon is digitized, the user is asked to enter a polygon description and polygon coordinate code. The codes are used as follows.

1 Polygon (x,y) relate to global (X,Y) coordinates
2 Polygon (x,y) relate to global (X,Z) coordinates
3 Polygon (x,y) relate to global (Y,Z) coordinates
4 Polygon (x,y) relate to grid (X,Y) coordinates

For example, if a polygon has a coordinate code of 2, and it is required to test to see whether an element is inside or outside the polygon, then the X and Z coordinates of the element will be used. This enables polygons to be used for plans, sections or even inclined grids.

The use of polygons is to be recommended over the array input available from D515, since layout changes are more easily accommodated by the latter method.

D809 Generate a test circular polygon

This is a program which was written specifically to generate the circular polygon used for checking the program against an analytical solution. It has been retained here in case similar tests need to be carried in the future. Normally, the user would not run this option.

D901 Print nodes and element in data file format

The user can use D901 to obtain a listing (formatted) of the global nodes and element/node definitions. This is often useful for debugging a mesh preparation error. For example, if it is known that there are errors in elements 34 and 45, which are connected to nodes 12,15,23,56,78,34 and 108, then these selected elements and nodes can be conveniently printed onto a single page for detailed checking.

D902 As for D901, with reversed element ordering

This module enables the user to obtain a printout/data file in which the node numbers for each element have been reversed. This is useful when the user has spent many hours numbers all nodes back to front! However, if you use it, then extra care should exercised to check that final nodes for each element are correct. At present it has only been tested for elements containing 3 or 4 nodes. It will not work for elements containing 8 or 9 nodes (ie curved elements).

D903 Display/modify number of nodes, benchmarks and elements

This module enables the user to adjust the numbers of elements, nodes and benchmarks. It is particularly useful, when one wishes to remove some benchmarks out of the database.

D904 Set automatic input files on/off

D905 Load in a new message file

The message file (MESAGE.BE) is a direct access file which is used by the menu drivers and many modules and it contains the menu and many prompt descriptions. This same prompts are contained in

an ASCII file (MESSB) which the user can edit using standard text editors. Provided that care is exercised to ensure that line numbers for each message are not altered, then the user may edit the ASCII message file and then update the 'system' message file in subdirectory C:\BEAPM\ using D905.

In this manner, much of the program could be translated into French for example. Alternatively, any prompt which the user finds misleading can be modified.

D906 Access/display selected messages

This enables the user to modify a few specific messages directly within the message file MESAGE.DD. If a particular message is found by the user to be mis-leading, then that message can be modified using this module. Obviously, care must be exercised to ensure that the message file is not corrupted in this manner by entering invalid messages.

D907 Print out selected messages

This will enable a user to printout some or all of the messages together with their corresponding message numbers, for correction or editing using D506.

D908 Set compact of full print mode

If compact print mode is set, then the form feed is suppressed from all printouts. This can save the volume of printout in some cases.

5.0 RUNNING BEAP

People using BEAP-DD for the first time will run modules via the menu driver. However, as more familiarity is gained, it usually becomes easier to run modules directly from DOS or as part of batch files. The following is an example of a run scenario for a typical underground problem.

5.1 EXAMPLE 1

Define mesh on paper and read off all coordinates of all grid corners.

List node coordinates and element/node definitions on paper

Capture node and element info into **D201** using a text editor. Although the filename is referred top as d201 here, remember that any file name may be used. Be sure to make full use of the 16 character description allowed for each lump. This is VERY useful for later on in the program. The small extra effort here to carefully name elements is worth it!

M101 Initialize the entire database.

D102 Initialize the polygon master file.

D201 Read in file D201.

Often there will be a data error here for a first time run. In this case, correct the data file and try again (ie M101 and D201)

D301 If an array of lumps are required, then run D301 to generate the extra lumps.

D204 Obtain an ASCII listing of newly generated nodal coordinates and lump / node numbering. The printout would be to print file TEMP.LST.

It is then advised to edit this printout of the newly generated nodes and lumps for updating back into the D201 type file. Assume this second file is called D201A. Its format would be similar to that of D201 prepared above, except that would contain all the lumps for the run, as opposed to only the basic outline of a grid.

If the mine layout was to be entered using polygons, then various polygon modules would then need to be run.

D102 Initialize the polygon master file.

D800 Digitizer menu

Option 1	Set up digitizer
Option 2	Digitize polygons

D206 Obtain a summary of polygon numbers and descriptions. This is important for the preparation of file D505 to follow.

D402 The polygons should be plotted for checking and for ready identification.

Create files **D202** and **D203**, **D207**, **D503** and **D505** according to the documentation. Often these will be created by copying these files from a previous run, and merely editing the relevant changes for the new run. In preparing file D505, polygon numbers would be obtained from the summary polygon print obtained from D206 above.

At this stage, a complete set of data files would then be available. If the polygon master file is correct and checked, then take a backup of this file (BEMPOL.BE) so that all polygons are not lost if M101 is run again.

If D201 has completed successfully, then run **D202** to read file D202 and carry out basic checks. Closely observe the displayed messages. If necessary, printout the file TEMP.LST to obtain a hard copy of these warnings. D204 may also be used to print out selected problem elements. Make corrections to the input data files and try again until all warnings have been eliminated or reasonably explained.

D203 Read in the material properties for each lump.

D207 Read in benchmarks and primitive stresses etc as prepared in the data file D207 above.

D208 Print out the header page.

D401 Run this module now to prepare a plot file for the mesh. It may also be necessary to run D207 here with a small data file containing the SSW and ISSW keywords to set the character size for the plots.

Use a plotter/display program to **view the plotfiles** (BEMMESH.DD and possibly BEMMESHE.DD and BEMMESHN.DD) on the screen or plotter. (See separate section on plotfiles).

D501 Calculate lumping maps. Examine the print file TEMP.LST to see whether there is sufficient disk space to store the influence coefficient matrix (BEMCO.DD).

D502 Calculate influence coefficients. This module can be run here or just before D507 is run to solve the equations. The latter option is preferable if a batch run is being planned for solution of the equations.

D503 Initialize layout for mining step zero. This program would read in data file D503 and set the initial mine layout prior to step number 1. For example, fault elements would be initialized as such, and reef and seam elements would be initialized as un-mined or solid.

D504 Increment mining step number by 1

D505 Set layout for current mining step. Here, file D505 would be read and groups of polygons would be processed against specified polygons to set mined / unmined areas in the model for the current step.

D506 Define boundary conditions for current mining step. Primitive stresses as specified by the PRIM1 and PRIM0 keywords would be applied to all mined elements.

D605 Obtain a printout of the mine layout to ensure that the polygons and groups of elements specified in file D505 have been correctly processed.

Once the various input data files have been checked, then use **D204** to obtain a hard copy of the input element geometry. This is an alternative to D605 above.

D701 One could also run **D701** to prepare a plot of the mine layout

Then use a text editor to create a batch file (eg RUN.BAT) which will run all the number crunching modules at once). Typically, this would be as follows

```
D507        Solve                                   equations
COPY                            temp.lst            lpt1
D508        Calculate                               benchmarks
COPY                            temp.lst            lpt1
```

D604 Print selected element results for selected lumps

Examine the printout to see whether the equation solution process (D507) has converged, and whether the remainder of printout from D508 appears reasonable. Do not take it for granted that if the program has run and printed results that everything is correct. Check the benchmark printout to see that no benchmarks are closer to an element than was expected. (the closest separation between each benchmark and the closest element is printed). The more you check the results, the better will be your understanding of what is happening in the model.

It may be necessary to run **D207** here to adjust plot parameters such as number of decimals and character size. (Use keywords ISSW and SSW here).

D701 prepare a plotfiles of selected element results.

BEMPOST Perform various runs to check benchmark values for failure subject to a selected criterion or to prepare plot files.

5.2 EXAMPLE 2

A second example is provided below as an alternative to the above example. It should demonstrate that there is flexibility in the precise manner and sequence in which the program modules are run

Running the program can be divided into three main areas.

- Data preparation and pre-processing
- Number crunching
- Result printing/plotting/post-processing

Often the pre- and post-processing will be run interactively, whist the number crunching is run in batch mode. The running of the programs is well demonstrated by means of a sample "session":-

PRE-PROCESSING

M101	Initialize a new project
D102	Initialize polygon data base
D801	Set up digitizer
D802	Digitize polygons

D206	Print polygons
D402	Plot polygons
-	Prepare data file D201
D201	Read in global nodes and lumps (D201)
D301	Generate more (rows and columns of) lumps
D204	Summary print of lump information
-	Edit errors in D201
D201	Re-run D201 - Read global nodes and lumps
D301	Re-run D301 - Generate more lumps
D204	Summary print of lump information
D401	Plot selected lump out-lines
-	Prepare data files D202, D203 and D207
D202	Read in lump groupings (+ check geometry)
D203	Read in lump properties
D207	Read in benchmarks, primitive stresses etc
D208	Print out header page
D908	Set printout verbosity to 1 = Reduced
D904	Switch to batch mode prior to number crunching

NUMBER CRUNCHING

D501	Calculate lumping maps
D502	Calculate influence coefficients
D503	Initialize layout for mining step zero
D504	Increment mining step number by 1
D505	Set layout for current mining step
D506	Define boundary conditions for current mining step
D507	Solve equations for current mining step
D508	Calculate benchmarks for current mining step
D509	Calculate energy release rates for current mining step
D510	Copy current step into multi-step storage
D511	Copy benchmarks into multi-step storage

Then D504 to D511 is repeated for each mining step to be modelled.

RESULT PRINTING, POST PROCESSING

D904	Switch to interactive or menu mode for post processing
D601	Define current step number to be processed
D602	Load current step from multi-step storage
D603	Load benchmarks from multi-step storage
D604	Print selected element results for selected lumps
D605	Print out mine layout
D606	Print selected benchmarks
D607	Prepare selected data for PC-MINE
D608	Prepare a spreadsheet file
D701	Prepare a cross plot for selected lumps for a selected item
D601	Change mining step
D602	Load current mining step
	Etc

5.3 EXAMPLE 3

A final example is presented in which input and number crunching as well as some result printing and processing would be carried out from a pre-prepared set of data files.

- Prepare data file D201
- Prepare data files D202, D203 and D207
- Prepare data files D503 and D505

M101	Initialize a new project
D201	Read in global nodes and lumps (D201)
D202	Read in lump groupings (+ check geometry)
D203	Read in lump properties
D207	Read in benchmarks, primitive stresses etc
D208	Print out header page
D501	Calculate lumping maps
D502	Calculate influence coefficients
D503	Initialize layout for mining step zero
D504	Increment mining step number by 1
D505	Set layout for current mining step
D506	Define boundary conditions for current mining step
D507	Solve equations for current mining step
D508	Calculate benchmarks for current mining step
D604	Print selected element results for selected lumps
D605	Print out mine layout
D701	Prepare a cross plot for selected lumps for a selected item
BEMPOST	Post processing of benchmark results

6.0 REFERENCES

1 Deist, F H and Georgiadis, E, A computer system for three-dimensional elastic analysis using a boundary element approach. Chamber of Mines of South Africa, Research Report No 43/76, Project No. GS1510 (1976).

2 Hocking, G, Development and Application of Boundary Integral and Rigid Block Methods for Geotechnics. PhD Thesis, University of London, UK (1977).

3 Watson, J O and Cowling, R, Application of three-dimensional boundary element method to modelling of large mining excavations at depth. 5th Int Conf on Num Meth in Geomechanics, Nagoya (1985).

4 Diering, J A C, Further Developments of the Boundary Element Method with Applications in Mining. MSc Thesis, University of the Witwatersrand (1982).

5 Lachat, J C, A further Development of the Boundary Integral Technique for Elastostatics. PhD Thesis, University of Southampton, UK (1975).

6 Lachat, J C and Watson, J O, Effective Numerical Treatment of Boundary Integral Equations : A formulation for three-dimensional elastostatics. Int Jnl for Num Meth in Eng, 10, (1976) 991 - 1005.

7 Bannerjee, P K and Butterfield, R, Boundary Element Methods in Engineering Science. McGraw Hill (1982).

8 Watson, J O, Hermitian Cubic boundary elements for plane problems of fracture mechanics. Int Jnl of Struct Mech and Materials Sc. Ed J Gittus (1982).

9 Diering, J A C, Advanced elastic analysis of complex mine excavations using three-dimensional boundary element techniques. PhD Thesis Pretoria University, South Africa (1987).

10 Cruse, T A, Numerical solutions in three-dimensional elastostatics. Int Jnl Solids and Structures, 5, (1969) 1259 - 1274.

11 Brebbia, C A. The Boundary Element Method for Engineers. Pentech Press (1980).

12 Diering, J A C and Stacey, T R. Three-dimensional stress analysis : A practical planning tool for mining problems. Proc 20th Int Symp APCOM, Johannesburg, SAIMM, 1, (1987) 33 - 42.

13 Hoek, E and Brown, E T, Underground excavations in rock, IMM, London (1982).

14 Diering, J A C and Yu, Y S. Three-dimensional stress analysis : A formulation for mining problems - Special issue on Mining Vol 5 No 2 Elsevier Appl Sc Publ (1988).

APPENDIX 1 -- EXAMPLE DATA FILES

Data file for shallow dipping reef (File dipreef.201)

```
8,101
0,0,0
1,250,250,-60
2,730,250,-200
3,730,750,-200
4,250,750,-60
5,0,0,0
6,1000,0,0
7,1000,1000,0
8,0,1000,0
1,4,1,2,3,4
REEF PLANE
101,4,5,6,7,8
SURFACE
```

Data file for shallow dipping reef (File dipreef.202)

```
  1,100,1,1,1,1,1,3,2,0
101,200,5,1,2,1,2,3,1,0
0,0,0,0,0,0,0,0,0,0
```

Data file for shallow dipping reef (File dipreef.203)

```
1,200,1.0,85,1,.2,.002
5000,3000,4,35,2,.5,50,2,.02,.1
0,0,0,0,0,0,0
```

Data file for shallow dipping reef (File dipreef.207)

```
HEADING
Shallow dipping reef example
PRIM1
-0.0540,0,0
0,-0.0405,0
0,0,-0.027
ITERATIONS
30,1.0,0.001,5,1.35
INTEGRATION
4,2,1.2,0.8,0.5,0.4
0.75
SSW
6,1.6
0,0
ISSW
11,0
0,0
SUBS
1
1,30000,0.25,1,0
BENCH
1,-40,500,0
SURF TEST 1
2,500,-40,0
SURF TEST 2
3,250,250,-30
BELOW NOD1
4,250,500,-30
DEPTH 30
5,500,500,-120
CENTRE -130
6,500,500,-100
CENTRE -100
7,500,500,-70
CENTRE -70
8,500,500,-40
CENTRE -40
-99,0,0,0
END
```

Data file for shallow dipping reef (File dipreef.503)

```
0,0,0,1,256,2
```

Data file for shallow dipping reef (File dipreef.505)

```
1,0,1,0,0,1
1,0,2,0,0,1
1,0,0,81,90,2
2,0,0,101,200,2
```

Vertical fault adjacent to reef File : Fault.201

18,10
0,0,0
1,0,0,0
2,20,0,0
3,20,0,20
4,0,20,0
5,20,20,0
6,20,20,20
7,0,40,0
8,20,40,0
9,20,40,20
10,0,60,0
11,20,60,0
12,20,60,20
13,0,80,0
14,20,80,0
15,20,80,20
16,0,100,0
17,20,100,0
18,20,100,20
1,4,1,2,5,4
BASE EDGE
2,4,2,3,6,5
SIDE EDGE
3,4,4,5,8,7
BASE RIGHT
4,4,5,6,9,8
SIDE RIGHT
5,4,7,8,11,10
BASE CENTRE
6,4,8,9,12,11
SIDE CENTRE
7,4,10,11,14,13
BASE LEFT
8,4,11,12,15,14
SIDE LEFT
9,4,13,14,17,16
BASE EDGE
10,4,14,15,18,17
SIDE EDGE

Vertical fault adjacent to reef File : Fault.202

```
1,1,1,1,1,1,1,3,0,0
2,2,2,1,1,1,2,3,0,0
3,3,1,1,1,1,1,3,0,0
4,4,2,1,1,1,2,3,0,0
5,5,1,1,1,1,1,3,0,0
6,6,2,1,1,1,2,3,0,0
7,7,1,1,1,1,1,3,0,0
8,8,2,1,1,1,2,3,0,0
9,9,1,1,1,1,1,3,0,0
10,10,2,1,1,1,2,3,0,0
0,0,0,0,0,0,0,0,0,0
```

Vertical fault adjacent to reef File : Fault.203

```
1,1,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
2,2,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
3,3,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
4,4,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
5,5,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
6,6,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
7,7,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
8,8,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
9,9,5.0,85,1,.2,.002,MINED ELEMENTS
100,50,25,5,.1,.5,50,2,.02,.1
10,10,0.1,85,1,.2,.002,FAULT ELEMENTS
1000,500,.25000,5,.1,.5,50,2,.02,.1
0,0,0,0,0,0,0
```

Vertical fault adjacent to reef File : Fault.207

```
HEADING
FAULT RUN
SUBS
1
1,100,.3,1,0
PRIM0
+10,0,0,0, +10,0,0,0, +5
ITERATIONS
30,1.2,0.001,4,1.20
BENCH
1 1.8 50 4.7
1
2 5.4 50 4.10
2
3 9.00 50 3.50
3
4 12.60 50 2.90
4
5 16.20 50 2.30
5
-99 0 0 0
END
```